

AD-A257 287



MENTATION PAGE

Form Approved
OPM No. 0704-0188

to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, completing and reviewing the collection of information, and sending the information to the Office of Management and Budget, Washington, DC 20503.

4. REPORT DATE June 1991		3. REPORT TYPE AND DATES COVERED	
4. TITLE AND SUBTITLE The Transformation and Representation of GIS Map Data for Use in Spatial and Temporal Problem Solving: A Case Study		5. FUNDING NUMBERS	
6. AUTHOR(S) Chubb, Douglas Walter J.		8. PERFORMING ORGANIZATION REPORT NUMBER 92-TRF- 0052	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army CECOM Signals Warfare Directorate Vint Hill Farms Station Warrenton, VA 22186-5100		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) DTIC ELECTE OCT 08 1992 S B D		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
1. SUPPLEMENTARY NOTES Presented at the 29th Annual Technical Symposium, Washington DC Chapter, Association for Computing Machinery - "Advanced Information Interfaces: Making Data Accessible", 19-20 June 1991			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Statement A: Approved for Public Release; Distribution Unlimited.		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Spatial and temporal problem solving within real-world domains is a contextually rich process which normally involves the processing of very large Geographic Information System (GIS) map databases. The U.S. Army is interested in these problems because they form the kernel of a tactical automated threat assessment system. This paper is a case-study describing the research and subsequent development of a new set of functions which transform raster-formatted GIS spatial data into a provably robust and economical form. This new representation has provided the basis for the development of an automated, multiagent, spatial and temporal problem solver. Prior to the development of this representation, and based upon studies of human problem solvers, it appeared that an automated problem solver would have to represent and process very large amount of GIS map data while solving spatial and temporal problems. A detailed mathematical analysis of the human problem solving process, however, resulted in the development of an automated system which has very modest memory requirements. The author argues that data representation issues cannot be properly considered until deep theoretical insight into the problem solving process has been developed; and this principle maybe extended to other domains.			
14. SUBJECT TERMS Artificial Intelligence, Problem Solving, Data Fusion, Automated Problem Solving, Geographic Information		15. NUMBER OF PAGES 7	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED		18. PRICE CODE	
18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED		19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	
20. LIMITATION OF ABSTRACT UL			

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to stay within the lines to meet optical scanning requirements.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	PR - Project
G - Grant	TA - Task
PE - Program Element	WU - Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - DOD - Leave blank.
DOE - DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - NASA - Leave blank.
NTIS - NTIS - Leave blank.

Block 13. Abstract. Include a brief (Maximum 200 words) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (NTIS only).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

THE TRANSFORMATION AND REPRESENTATION OF GIS MAP DATA FOR USE IN SPATIAL AND TEMPORAL PROBLEM SOLVING: A CASE STUDY

Douglas Walter. J. Chubb

U.S. Army CECOM Center for Signals Warfare, Vint Hill Farms Station
Warrenton, Virginia 22186-5100

Abstract

Spatial and temporal problem solving within real-world domains is a contextually rich process which normally involves the processing of very large Geographic Information System (GIS) map data bases. The U.S. Army CECOM Center for Signals Warfare is interested in these problems because they form the kernel of a tactical automated threat assessment system. This paper is a case-study describing the research and subsequent development of a new set of functions which transform raster-formatted GIS spatial data into a provably robust and economical form. This new representation has provided the basis for the development of an automated, multi-agent, spatial and temporal problem solver. Prior to the development of this representation, and based upon studies of human problem solvers, it appeared that an automated problem solver would have to represent and process very large amounts of GIS map data while solving spatial and temporal problems. A detailed mathematical analysis of the human problem solving process, however, resulted in the development of an automated system which has very modest memory requirements. The author argues that data representation issues can not be properly considered until deep theoretical insight into the problem solving process has been developed. The author suggests that this principle may be extended to other problem domains.

I. Introduction

In the Fall of 1984 the U. S. Army CECOM Center for Signals Warfare (C2SW) began a basic research effort [10] to investigate the possibility of developing an automated system which would assist the U.S. Army Division Level

Intelligence Officer (G2) and Staff during their preparation of an enemy tactical threat assessment report. The report consists of the G2's predictions for the next three or four hours of the most likely tactics the observed enemy forces will pursue. Early on in this research, the author conducted a number of in-depth interviews with several G2s and Staff members. It quickly became apparent that the threat analysis process (as practiced by humans) is non-deterministic and is properly characterized as a "hypothesize-and-test" paradigm [2, 8]. During threat assessment, the G2 generates a number of likely enemy tactical hypotheses. Tactic generation is based upon the G2's past experiences with the enemy, their location within the tactical domain, and their presumed tactical objective(s). Each tactical hypothesis is then evaluated by the G2 using a *gestalt-like* process whereby enemy forces are imagined performing the hypothetical tactic within the domain. Potential difficulties believed to occur during tactic evolution within the domain are noted and the feasibility of the tactic is measured. Those tactics which appear to be the most feasible, efficacious, and logically consistent with the perceived long-term enemy mission become the probable threat.

The accuracy of the G2's hypothesis evaluation process clearly depends upon his knowledge of the terrain and weather conditions and the presence of other friendly or enemy forces within the domain and their effect upon the evolution of the hypothetical tactic. C2SW's research goal has been to develop an automated threat assessment system which can generate and evaluate tactical hypotheses for the G2, while leaving the selection of the most likely threat scenario to the human user. For practical use by the Army G2, the threat assessment paradigm must be *domain independent* (performance not affected by choice

of tactical domain), and *extensible* to any mix of opposing enemy forces.

This paper describes the development and implementation of a tactical hypothesis evaluation algorithm which has become an integral part of a threat assessment system being developed at C2SW. In particular, this paper focuses on the development of a spatial transformation and representation which has made it possible for the system to comply with the stated requirements of domain independent and force extensibility. We will show that a careful analysis of the G2 problem solving process has resulted in a mathematical reformulation of the problem which has drastically reduced both the complexity of the problem solving process and its associated spatial data requirements. The author suggests that a similar approach may be extended to other problem domains.

II. Hypothesis Evaluation

2.1 Data Representation Requirements

A tactical hypothesis to be evaluated is represented as the list,

$$\{F^k_{(i,j)}, \{F^h_{(k,l)}, \dots, F^d_{(s,f)}\}, \{G^k_{(a,t)}, \dots, G^d_{(d,q)}\}, T\} \quad (1)$$

where $F^k_{(i,j)}$ represents a type k (e.g., artillery, armor) military unit or force, F^k , located at map coordinate location (i,j) whose tactic is being evaluated. The second element of the list, $\{F^h_{(k,l)}, \dots, F^d_{(s,f)}\}$, represents the list of other known (enemy and friendly) forces and their respective locations within the domain. The third element, $\{G^h_{(a,t)}, \dots, G^d_{(d,q)}\}$, represents the locations of a set of tactical objective(s) or goal(s) for each force mentioned in (1). Lastly, T is an optional variable which represents the hypothesis time constraint. If T is present the hypothesis is called a *temporal problem*, else a *spatial problem*. As an example, a temporal hypothesis might assume that an enemy force F^k is located at coordinate (i,j) and ask how might F^k move to goal location $G^k_{(a,t)}$ in less than $T=30$ minutes? In order to properly evaluate this hypothesis a great deal of *a priori* information about F^k is necessary. This also includes a characterization of

F^k 's mobility and logistical requirements and how F^k 's tactics might be effected by the presence of other F^y within the domain. Note that in general equation (1) is but one of a set of equations which must be evaluated in parallel. Associated with each F^k is (possibly) an associated set of F^y each of which are presumed to have their own goals which they wish to satisfy. Thus the evaluation process is inherently parallel.

Since most of the forces of interest are land-mobile, a tactical hypothesis evaluation system (THES) must be able to quickly access and process the terrain data associated with the hypothetical evolution of the tactic in time. In particular, an Army Divisional area of interest represents 1.5×10^{10} meters². If M is an Army map of this area and X is an $n \times n$ matrix (raster formatted) covering of M , then associated with every element, x_{ij} , belonging to X , is an associated spatial area within M , for $i,j=1,\dots,n$. Also associated with each x_{ij} is a set of $k > 0$ spatial features, $\{f_1, \dots, f_k\}_{ij}$, and feature values ($0 \leq f_i < r$, $i=1,\dots,k$) which describe the (syntactic) features found within each associated area of M . If each x_{ij} represents 10 meters² within M and the features associated with each 10 meters² are described using 8 bytes of data then X will require 10^{11} bytes to represent the syntactic features in M .

Semantic map information should also be represented and saved within X . Associative syntactic information, such as the locations of tactical targets, and spatial and functional relationships between objects such as roads and bridges, for example, should be identified and saved as part of X 's semantic representation. However, determining *what* information to represent is a function of each F^k . Semantic associations important to F^k are not necessarily important to another F^d . Semantic associations may also vary depending upon the *actions* being executed by the F^k . For example, F^k may consider the secondary road xyz useful during a retreat, but unacceptable for use during an attack. Finally, syntactic map features, which are a constituent part of the semantic expression, can change abruptly during a tactical engagement. Therefore, the THES semantic representation must be *provably robust*. That is, any change to

any feature value in X must produce a deterministic change to X 's semantic data structure.

In summary, the THES data base, X , is required to represent,

- 1) syntactic map information,
- 2) semantic information (to be determined) which will be somehow associated with each force F^k , and,
- 3) semantic information in a provably robust form.

In addition, there are two implicit THES requirements which may effect the representation of X ,

- 1) that the THES process tactical hypotheses in parallel,
- and,
- 2) that knowledge-bases associated with each F^k must be accessible to the THES process.

2.2 A Spatial Data Transformation

Fortunately, the difficulties involved in developing a THES data base as described forced the author to reconsider the hypothesis evaluation problem statement, (1). Additional attempts to further abstract the THES process resulted in an important reformulation of the process model. Using Schank-like notation, it was noted that each force F^k could be thought of as an *actor*, AC , performing an *action*, A , within a *domain*, C . If we assume that actor action execution is reasoned and purposeful, then the tuple expression $\{AC, A, C\}$ represents the domain situation, from the actor's perspective, just prior to the execution of the action. C represents the set of features and feature values known to the actor which describe the domain. In particular, the actor "believes" that C contains features which make the *initiation of action execution possible*. We will assume that the actor *believes that he knows* what conditions are necessary to make action execution possible [5]. Action execution, then, becomes an actor's reasoned context-modification paradigm which either makes the execution of another action possible or produces the desired goal state within C . Actor goals are represented by a set of features and features values in C . If E is the action execution function, then,

$$E\{AC, A, C\}_i = \{C_{i+1}\} \quad (2)$$

where C_{i+1} is a new context expression. In general, context expressions may contain other actors (FY) and hence represent all of the ingredients found within the THES force-location portion(s) of (1). The temporal evolution of a single-actor THES expression is represented as the execution of a time-contiguous sequence of tuples, where,

$$P = \{\{AC, A, C\}_1, \{AC, A, C\}_2, \dots, \{AC, A, C\}_k\} \quad (3)$$

and C_{k+1} contains the actor's goal. The sequence, P , is frequently called a *plan* [1, 4].

Equations (2) and (3) help to describe the underlying relationships between the tactical actor, the domain, and action execution. In particular, (2) suggests that domain expressions can be partitioned into two mutually exclusive sets: the set of features and associated feature values which *effect* the execution of an intended action, and the complement of this set (which do not effect action execution). Likewise, the former set can be partitioned into two mutually exclusive sets: the set of features and feature values which *make action execution possible*, and the set C' of features and feature values which make *action execution impossible*. Given an actor and an intended action, the domain expression C_i may be partitioned into mutually exclusive subsets, i.e., $C_i = C_i' \cup \sim C_i'$. Fortunately, for most tactical equipment, a description of the elements of C_i' is readily available [9]. Tactical equipment is tested extensively and subjected to a wide range of environment conditions during its development. The resulting equipment operational capabilities are parameterized as a function of equipment type (AC) and function (A). Provided with the tuple $\{AC, A_i\}$, the C_i' expression may be developed. Once developed, we define the function FBD_T as follows,

$$FBD_T(X) \rightarrow \{1,0\}$$

where,

$$FBD_T(x_{ij}) = 0 \text{ iff } C' \cap \{f_1, \dots, f_k\}_{ij} = \emptyset$$

else,

$$FBD_T(x_{ij}) = 1 \quad (4)$$

for all x_{ij} in X , $i, j = 1, \dots, n$.

FBD_T is called a *Functional Binary Decomposition Transformation* since it functionally relates action execution to the elements of C , and transforms X into a binary representation. Since $X \approx C$, within the accuracy and resolution limitations of $\{f_1, \dots, f_k\}_{ij}$, we will use both expressions equivalently and frequently write $FBD_T(C)$. Strictly speaking, however, X is a finite approximation of C .

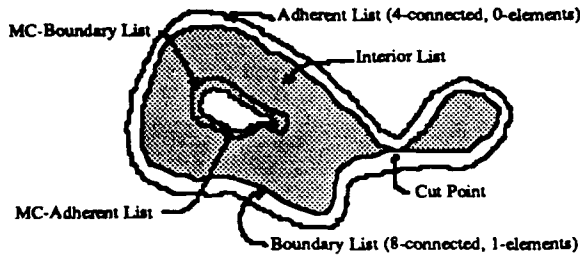


Figure 1

An FBD region's boundary lists, adherent lists, and cut points.

Since $FBD_T(X) \rightarrow \{1,0\}$, the range of the function (an $n \times n$ binary array of general complexity) can be completely characterized by the set of all x_{ij} in X such that $FBD_T(x_{ij}) = 1$ (everything else in the range equals 0). We define this set of *1-element* x_{ij} to be 8-connected while its complement set of *0-element* x_{ij} is 4-connected [11]. Each path-connected set of 8-connected 1-elements is called a *region*. A region is characterized by its: 1) boundary list, 2) adherent list, and 3) cut-points [7]. Since a region may be multiply-connected (MC), MC-boundary and MC-adherent lists are saved as well as those cut points associated with these lists. The transformation from $FBD_T(X)$ to a set of regions, $\{fbd_1, fbd_2, \dots, fbd_k\}$, is called a *Functional Binary Decomposition Representation*, FBD_R . That is,

$$FBD_R \circ FBD_T(C) = \{fbd_1, fbd_2, \dots, fbd_k\} \quad (5)$$

where each fbd_i is a region description, i.e., boundary lists, adherent lists, and cut-points (see Figure 1).

Chubb and Lavine [7] have proved that the set of fbd_i , which we will refer to as the FBD representation, is *robust*. We will now describe how this representation can be used to solve single actor THES spatial and temporal problems.

2.3 Single Actor Spatial Problem Solving.

As discussed previously, the general spatial problem hypothesis statement is: Can AC move from coordinate point (a,b) to point (y,z) ? Because spatial problems do not consider time an hypothesis parameter, the domain feature values are assumed to be invariant during the evaluation process. Since the 0-elements of the FBD representation represent areas where action execution (i.e., mobility) is possible, we reformulate the problem to ascertain if 0-element points $x_{a,b}$ and $x_{y,z}$ are *path-connected*. Path-connectedness can be proved using a number of admissible algorithms, notably the A* heuristic search or similar region growing algorithms [8]. A more efficient algorithm which makes direct use of the FBD representation is the Straight Line Path Algorithm (SLPA) [6]. The SLPA makes use of the fact that each region's *adherent list* "encircles" that region. The SLPA is described as follows. Point (a,b) is the actor's "start point"; point (y,z) is the actor's goal location.

SLPA Start (a,b) & (y,z) :

PUSH arguments (a,b) & (y,z) to SLPA-Stack.

A: POP SLPA-Stack list of arguments.

Build straight-line path between x_{ab} & x_{yz} ?

If yes:

PUSH (a,b) & (y,z) onto path-pieces-stack.

Check SLPA-Stack:

If empty:

RECONSTRUCT path-pieces-stack.

Done.

Else, GOTO A.

If no:

Path is occluded by Region R_i .

Intersection of path and R_i 's Adherent List is 0-element point x_{cd}

B: BRANCH CLOCK-WISE about R_i 's

Adherent List from (c,d) .

For each point, (c', d') :

Build straight-line path between $x_{c'd'}$ & x_{yz} ?

If yes: PUSH (c', d') & (y,z)

GOTO A.

Else: GOTO B.

C: BRANCH COUNTER-CLOCK-WISE about R_i 's Adherent List from (c,d)

For each point, (c'', d'') :

Build straight-line path between $x_{c''d''}$ & x_{yz} ?

If yes: PUSH (c'', d'') & (y,z)

GOTO A.

Else: GOTO C.

The C2SW implemented *Common LISP* version of SLPA includes a (region) check at branch points B and C to ensure that program loops don't occur. The path-pieces-stack contains a list of region-to-region straight-line path segments. Path segments are represented by their start and stop points. Since, in general, these points will also be members of region adherent lists, the regions and the path segments connecting regions can be computed. RECONSTRUCT builds a highly abstracted tree of path solutions from start point to goal point. Tree arcs represent straight-line path segments; tree nodes represent 1-element regions.

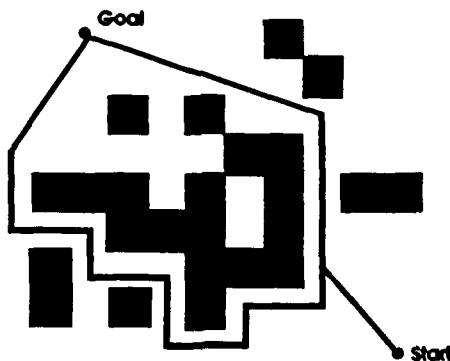


Figure 2.

A single-actor SLPA solution set. Black areas represent FBD 1-elements; white areas represent 0-elements. Region Adherent List information is used to branch CW and CCW about each region. Spatial features are assumed to be invariant during the solution process.

Because the *a priori* development of a real-world "best path" criteria is not possible [4], the SLPA develops a family of possible solutions, see Figure 2. The SLPA produces a minimal solution set which is mathematically *complete*, i.e., any additional path solution can be shown to be topologically equivalent to one and only one of the SLPA path solutions. Chubb [6] has shown that for most domains the SLPA's search space is significantly smaller than an equivalent A^* (using a shortest path length heuristic) search space.

In conclusion, the SLPA develops a single actor, spatially and temporally invariant set of path solutions. For multi-actor spatial problem solving, however, a representation will be

needed which will permit the introduction of feature value changes within C which have been caused by actor action execution as a function of time. Since all temporal problem solutions are spatial solutions, we will limit our discussion to multi-actor temporal problem solving.

2.4 Multiple Actor Temporal Problem Solving

A temporal problem solver will require some means of computing the amount of actor time required to transit any 0-element point belonging to $FBD_T(X)$. To provide this capability the FBD representation is augmented with an $n \times n$ array of real numbers, wgt_{ij} , such that,

$$K_{ij} = P_{size} + (V_{max} * wgt_{ij}) \quad (6)$$

where, K_{ij} = AC's transit time through x_{ij}
 P_{size} = spatial size (resolution) of x_{ij} in X
 V_{max} = AC's maximum action execution velocity

and, $0.0 < wgt_{ij} \leq 1.0$ for all $i, j = 1, \dots, n$.

The wgt_{ij} are called *coefficients of compliance* [3]. Each wgt_{ij} represents a measure of the expected *actor action execution compliance* for every element in $FBD_T(X)$. A compliance value of 1.0 implies that no features exist within the x_{ij} which will hinder the execution of the actor action. As such the computed K_{ij} value will be minimal. We assume that every FBD representation 0-element has some positive action compliance value. Conversely, one-elements are assigned a coefficient value of 0.0 and the computed K_{ij} value is undefined (infinite).

Given the following:

- a) an actor and an intended action,
- b) an *augmented* FBD representation,
- c) a start location (a, b) ,
- d) a goal location (y, z) ,

and, e) an action time constraint T, per (1), then we define a *temporal solution* to be the set S of all 0-elements belonging to $FBD_T(X)$ such that,

- 1) x_{ab} and x_{yz} are elements of S,
- and, 2) x_{rt} an element of S then there exists a 0-element path Z connecting points x_{ab}, x_{rt}, x_{yz} such that,

$$T \geq \sum_{i=ab}^{yz} K_i \quad (7)$$

where x_i is an element of Z for all $i=ab, \dots, rt, \dots, yz$, and $Z \subseteq S$, see Figure 3.

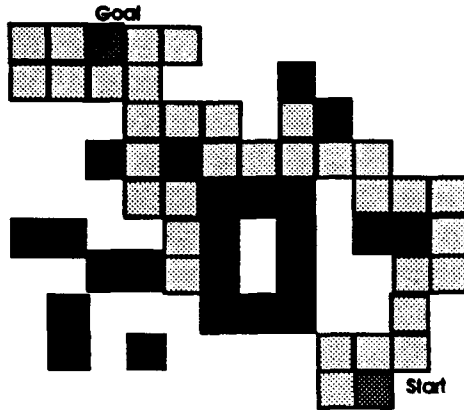


Figure 3.

Temporal solution set S . White points are FBD representation 0-elements; black points are 1-elements. Points belonging to S are gray. Dark gray points represent start and goal points x_{ab} and x_{yz}

The set S is developed using the following algorithm:

Start: (Build-S ab yz AC T)

O-LIST = {(a,b)}

A: SELECT Min-Path-Time from O-LIST

O-LIST empty?

If yes: GOTO C.

Else: Pick min point

GOTO B

B: POST on BLACKBOARD any actor action execution feature-changes, point, and time interval = [min-path-time, K_{ij}].

CHANGE actor features, as necessary, for time = min-path-time + K_{ij} .

BUILD-4-NEIGHBORHOOD about min point

Store min point TRANSIT-TIME on 4-neighbors

Compute Manhattan-Distance to (y,z)

Is min-point time + Manhattan-Distance + Transit Time > T?

If yes: GOTO A

Point = (y,z)?

If yes: Store point in S-List.

GOTO A

Else: Store point in O-list

GOTO A

C: Point (y,z) an element of S-List?

If no: "No Solution Possible!"

Stop.

Else: Using elements of S-List, build

min-time-path P_{min} from x_{ab} to x_{yz} .

D: S-List = P_{min} ?

If yes: S = S-List.
done.

Else:

FIND leaf nodes of P_{min}

Leaf node have min-path-time "children" belonging to P_{min} ?

E: If yes: Compute lattice min-path-time to element of P_{min}

Path-time > T?

If yes: Remove leaf node from S-List.

GOTO D.

Else: Link leaf-node as part of lattice to P_{min}

GOTO D.

F: Else: Compute min-path-time to point and to "parent" point.

Parent have "children" linked to P_{min} ?

If no: GOTO F.

Else: GOTO E

For each x_{ru} processed, Build-S (branch point B:) posts its min-path-time entry and transit times as the closed interval (t_1 t_2) on a *network blackboard* where,

$$t_1 = \sum_{i=ab}^{fg} K_i \quad (8)$$

and $t_2 = t_1 + K_{ru}$.

When action execution begins within x_{ru} , its domain features are assumed to be invariant. Once this level of processing is completed, AC posts any feature changes which have been produced by his action execution within x_{ru} on the blackboard. In addition, AC also checks the blackboard for any other feature changes which may have been posted by another actor for time t_2 . Domain feature value changes either produce an on-line change to the actor's FBD representation or a change in the value of an applicable wgt_{ij} . These representational changes are possible because the FBD representation is robust and because Build-S always posts an absolute time measurement. The absolute time measurement process permits a set of actors to be *asynchronously networked* together. Each actor within the network attempts to satisfy his own goals. Actor-to-actor interactions are developed indirectly through the blackboard by dynamically changing each actor's domain representation. This networking schema satisfies the THES requirements established in (1) as required.

erally orders of magnitude smaller than the associated GIS map data representation. Actor network interactions continue to be a research topic of interest.

References

- [1] Chapman, David, "Planning for Conjunctive Goals", MIT Tech Report #AI-TR-802, 1985.
- [2] Charniak, Eugene, Christopher K. Riesbeck, and Drew V. McDermott, , "Artificial Intelligence Programming", Lawrence Erlbaum Assoc, Hillsdale, New Jersey, 1980.
- [3] Chubb, Douglas W.J., "Spatial Problem Solving Using a Generalized N-Dimensional Binary Representation", Proc. 5th IEEE International Symposium of Intelligent Control 1990, Vol. I., pp. 605-610, 1990.
- [4] Chubb, Douglas W.J. "A Proof of Chapman's Conjecture for Tactical Conjunctive Plan Recognition", Proc. 1989 Tri-Service Data Fusion Symposium: Multiple Dimensions of the Data Fusion Process, Vol I., pp. 93-101, 1989.
- [5] Chubb, Douglas W.J., "Transitioning Mechanized Plan Recognition from Closed to Real-World Domains", Proc. SPIE Sensor Fusion II: Human and Machine Strategies, Vol. 1198, pp. 538-549, 1989.
- [6] Chubb, Douglas W.J., "An Introduction and Analysis of a Straight Line Path Algorithm for use in Binary Domains", Proc of the 1987 Workshop: Spatial Reasoning and Multi-Sensor Fusion, AAAI sponsored, pp. 220-229, 1987.
- [7] Chubb, Douglas W.J. and David Lavine, "Proof of Robustness of the Functional Binary Decomposed Semantic Spatial Representation", C2SW TR86-1, 1986.
- [8] Feigenbaum, Edward A., and Avron Barr, "The Handbook of Artificial Intelligence, Volume I", William Kaufman, Inc., Los Altos, California, 1981.
- [9] Jane's Military Vehicles and Ground Support Equipment, Seventh Edition, Christopher F. Foss & Terry J. Gander Editors, London, England, 1986.
- [10] Lavine, David "A Design for a Battlefield Situation Assessment System", Final Report, C2SW contract #DAAK21-84-C-0037, LNK Corporation, MD, 1985.
- [11] Rosenfeld, Azriel, and Avinash C. Kak, "Digital Picture Processing, Volume II", Academic Press, Orlando, Florida, 1982.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

SUPPLEMENTARY

INFORMATION

III. Conclusions

The augmented FBD representation meets all but one of the stated requirements for a THES data base. The FBD representation represents the relationship between the domain and actor action execution, *i.e.*, it is a spatial representation of plan tuple $\{AC, A, C\}_i$. This representation is intimately associated with each actor in a provably robust form. We have shown how this representation forms the basis of an actor-based evaluation process in accordance with equation (1) where actor action execution indirectly effects each of the other actors within the problem domain. Strictly speaking, however, the FBD representation does not meet the requirement to *represent syntactic map information*. The author maintains that this "requirement" is unnecessary and ill-founded.

The requirement to represent syntactic map information initially arose when the THES problem was first formulated. We reasoned that since most actors are land-mobile, "the THES must be able to quickly access and process the terrain data associated with the hypothetical evolution of the tactic in time" (section 2.1). The problem was that we assumed that *all* of the syntactic information was potentially associated with the evolution of a tactic. In support of this assumption, we noted that (human) G2 problem solvers make use of volumes of very detailed maps while solving tactical problems. We then erroneously concluded that if humans require numerous detailed maps while solving tactical problems, then any automated process less intelligent than a human problem solver **must minimally** have access to the same corpus of information. Hence the THES data base requirement to represent very large amounts of syntactic map information.

In retrospect, it now appears that the G2 uses maps to augment, as necessary, his knowledge base while solving spatial or temporal problems. However, it is unlikely that any G2 would require *all* of the map information during any problem solving session. That proper subset of map data used to solve a particular spatial/temporal problem can be described in retrospect by the G2 while solving a problem. The same subset, however, can be described *a priori* once the underlying theory of the G2 problem solving process is well understood. The theory

provides the only sure basis for the subsequent development of a series of rational decisions which must be made concerning the proper representation and usage of the THES data base.

Some of man's early attempts at powered flight produced machines which would (slowly) flap large mechanical "wings". Movies illustrating the disastrous results of these early experiments are quite humorous. Apparently the designers had reasoned that because birds fly by flapping their wings, men should do likewise. It was only after deep theoretical insight was developed into the dynamics of flight and airfoils in particular that the first practical aircraft were successfully developed. Frequently, problems which appear to be intractable have not yielded because the underlying theoretics of the problem are ill understood. Adequate research and inquiry should always precede problem formulation. This is especially true with automated systems which are capable of accessing and processing data very quickly. Unfortunately, high speed computers have frequently encouraged the development of quickly devised solutions which, in the short-term, may meet immediate system needs but which eventually precipitate a variety of system problems. Initial attempts to develop a THES data base were discouraging because the representation chosen was, like the early flying machines, based upon the faulty assumption that an existing methodology, *i.e.*, the flight of a bird or a human problem solver, should become the literal basis for the design of an automated process. Initially, the THES problem appeared to be formidable and unyielding because its GIS data base representation and access requirements seemingly could not be compromised. We had an existence "proof of principle": the human G2 problem solver. Unfortunately, too much was made of these early observations with G2s and we attempted to build a machine which "solved problems like a G2". Once the underlying THES principles were discovered and examined, however, solutions to the representational problem became self evident and immediate. The author suggests that this approach might be applicable to other similar problem domains.

The current *Common Lisp* implementation of the C2SW THES is provably domain independent, extensible, and robust, as required. Actor memory requirements are very modest and gen-